

In the Claims:

1. (Currently Amended) A method, ~~at least partially implemented by a~~
computing device, for processing a database query, comprising:

partially pre-aggregating records in a database to provide a result that
contains at least two records having like grouping column values;

aggregating records derived from the ~~partial pre-aggregation result that~~
~~contains at least two records having like grouping column values~~ to provide a
result that contains records having unique grouping column values; and

partially pre-aggregating the records in the database only if an estimation,
based ~~in part~~ on a calculation of a probability that a record will be absorbed by a
group of records already in memory, indicates that a number of records in the
result that contains at least two records having like grouping column values is
significantly less than a number of records in the database, ~~output records from the~~
~~partial pre-aggregation will be significantly less than a number of input records to~~
~~the partial pre-aggregation,~~ wherein the estimation is based on factors comprising:

a number of output records, T(N);

a number of input records, N; and

a relationship:

$$T(N) = M + (N - M)(1 - A(R(M))) = M + (N - M) \sum_{i=1}^D (1 - p_i)^{R(M)};$$

wherein M records can fit into memory.

1 2. (Original) The method as recited in claim 1, wherein the partially
2 pre-aggregating further comprises:

3 maintaining a record store in memory, the record store having one record
4 for each different grouping column value encountered in the operation;

5 receiving a new record;

6 combining the new record with a record having the same grouping column
7 value, if such a record exists; and

8 adding the new record to the record store in the memory if there is no record
9 in the record store that has the same grouping column value as the new record.

10
11 3. (Original) The method as recited in claim 2, further comprising:

12 adding additional new records to the record store until the record store
13 reaches a capacity such that it can accept no new records; and

14 outputting one or more records from the record store to a subsequent
15 database operator.

16
17 4. (Original) The method as recited in claim 3, wherein after the one or
18 more records have been output to the subsequent database operator, the adding and
19 outputting are repeated until there are no new records to process.

20
21 5. (Original) The method as recited in claim 4, wherein any records
22 remaining in the record store after there are no new records to process are output to
23 the subsequent database operator.

1 6. (Original) The method as recited in claim 3, wherein the subsequent
2 database operator is a join.

3
4 7. (Cancelled)

5
6 8. (Original) The method as recited in claim 1, wherein the partially
7 pre-aggregating includes utilizing a hashing function.

8
9 9. (Original) The method as recited in claim 1, wherein the partial pre-
10 aggregating creates a record store in memory, and wherein the method further
11 comprises utilizing the record store in memory for one or more other database
12 operators.

13
14 10. (Cancelled)

15
16 11. (Original) A computer programmed to perform the method recited in
17 claim 1.

18
19 12—23. (Cancelled)

24. (Currently Amended) A relational database computer program stored on a computer-readable medium, the relational database computer program comprising computer-executable instructions that, when executed on a computer, perform the following steps comprising:

receiving a stream of input records;

partially pre-aggregating the input records according to a single grouping column to provide a result that contains at least two records having like grouping column values, wherein the ~~partial pre-aggregation~~ partially pre-aggregating the input records is performed if an estimation, based in-part on a calculation of a probability that a record will be absorbed by a group of records already in memory, indicates that a number of ~~output records from the partial pre-aggregation will be significantly less than a number of input records to the partial pre-aggregation;~~ records in the result that contains at least two records having like grouping column values is significantly less than a number of records in the stream of input records, wherein the estimation is based on factors comprising:

a number of output records, T(N);

a number of input records, N; and

a relationship:

$$T(N) = M + (N - M)(1 - A(R(M))) = M + (N - M) \sum_{i=1}^D (1 - p_i)^{R(M)};$$

wherein M records can fit into memory;

joining the partially pre-aggregated records with other data to create a record store; and

aggregating records within the record store to provide a result that contains records having unique grouping column values.

1 25. (Original) The relational database computer program as recited in
2 claim 24, wherein:

3 the record store has a capacity that is less than the number of records in the
4 stream of input records; and

5 the aggregating each input record is performed until the record store reaches
6 capacity.

7
8 26. (Cancelled)

9
10 27. (Previously Presented) The method of claim 1, wherein the
11 estimation is based, in part, on an estimated absorption rate by which records are
12 absorbed by records in memory.

13
14 28. (Currently Amended) The method of claim 27, wherein the
15 absorption rate of ~~available memory~~ is estimated, in part, based on a number of
16 records expected to be processed.

17
18 29. (Previously Presented) The method of claim 28, wherein the number
19 of records expected to be processed is estimated, in part, based on a number of
20 records that will fit in memory.

21
22 30. (Cancelled)

23
24 31. (Currently Amended) The method of claim [[30]]1, wherein the
25 number of input records, N, is known.

1
2 32. (Currently Amended) The method of claim [[30]]1, wherein the
3 number of input records, N_i , is estimated.
4

5 33. (New) The relational database as recited in claim 24, wherein the
6 partially pre-aggregating further comprises:

7 maintaining a record store in memory, the record store having one record
8 for each different grouping column value encountered in the operation;

9 receiving a new record;

10 combining the new record with a record having the same grouping column
11 value, if such a record exists; and

12 adding the new record to the record store in the memory if there is no record
13 in the record store that has the same grouping column value as the new record.
14

15 34. (New) The relational database as recited in claim 33, wherein the
16 steps further comprise:

17 adding additional new records to the record store until the record store
18 reaches a capacity such that it can accept no new records; and

19 outputting one or more records from the record store to a subsequent
20 database operator.
21

22 35. (New) The relational database as recited in claim 34, wherein any
23 records remaining in the record store after there are no new records to process are
24 output to the subsequent database operator.
25

1 36. (New) The relational database as recited in claim 24, wherein the
2 partially pre-aggregating includes utilizing a hashing function.

3
4 37. (New) The relational database as recited in claim 24, wherein the
5 partial pre-aggregating creates a record store in memory, and wherein operation of
6 the relational database further comprises steps utilizing the record store in memory
7 for one or more other database operators.

8
9 38. (New) The relational database as recited in claim 24, wherein the
10 estimation is based, in part, on an estimated absorption rate by which records are
11 absorbed by records in memory.

12
13 39. (New) The relational database as recited in claim 38, wherein the
14 estimated absorption rate is estimated, in part, based on a number of records
15 expected to be processed.

16
17 40. (New) The relational database as recited in claim 39, wherein the
18 number of records expected to be processed is estimated, in part, based on a
19 number of records that will fit in memory.

20
21 41. (New) The relational database as recited in claim 24, wherein the
22 number of input records, N_i , is known.

23
24 42. (New) The relational database as recited in claim 24, wherein the
25 number of input records, N_i , is estimated.